Advanced Control

January 10, 2014

École Centrale Paris, Châtenay-Malabry, France

Anne Auger

INRIA Saclay – Ile-de-France

INVENTORS FOR THE DIGITAL WORLD

Dimo Brockhoff

INRIA Lille – Nord Europe

Course Overview

Date		Topic
Fri, 10.1.2014	DB	Introduction to Control, Examples of Advanced Control, Introduction to Fuzzy Logic
Fri, 17.1.2014	DB	Fuzzy Logic (cont'd), Introduction to Artificial Neural Networks
Fri, 24.1.2014	DB	Bio-inspired Optimization, discrete search spaces
Fri, 31.1.2014	AA	Continuous Optimization I
Fri, 7.2.2014	AA	Continuous Optimization II
break		
Fri, 28.2.2014	AA	The Traveling Salesperson Problem
Fr, 7.3.2014	DB	Controlling a Pole Cart
Fr, 14.3.2014		written exam (paper and computer)

all classes + exam at **8h00-11h15** (incl. a 15min break around 9h30) here in CTI-B3

The Exam

- Friday, 14th March 2014 from 08h00 till 11h15
- open book: take as much material as you want
- combination of
  - questions on paper (to be handed in)
  - practical exercises (send source code and results by e-mail)
- 2 ECTS points

All information also available at

http://researchers.lille.inria.fr/~brockhof/advancedcontrol/

(exercise sheets, lecture slides, additional information, links, ...)

Advanced Control: What is that?

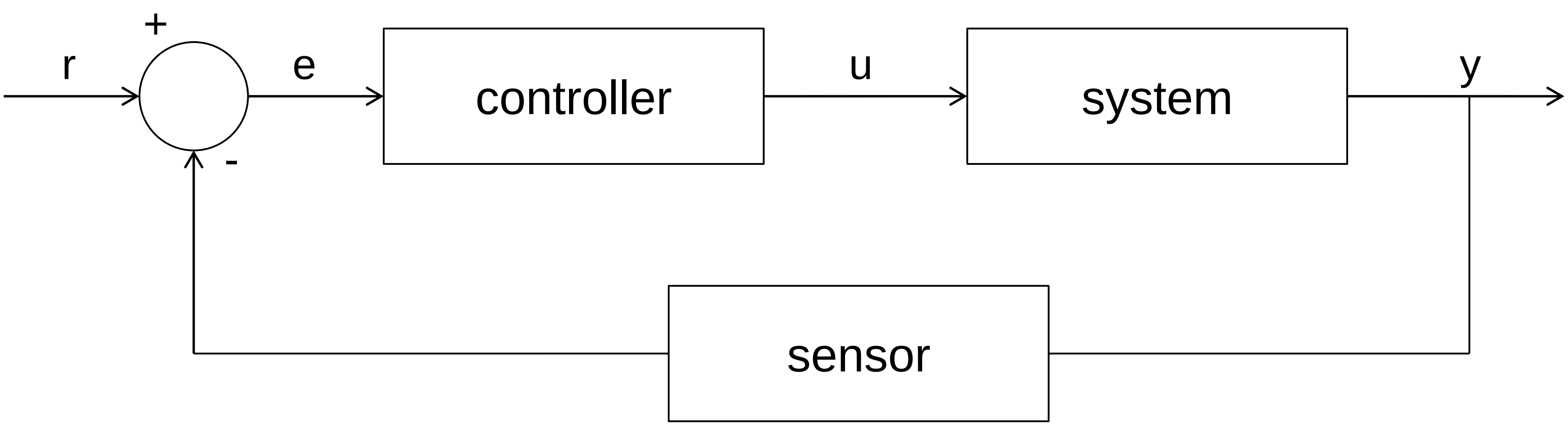
Advanced Control: What is that?

What is Control?

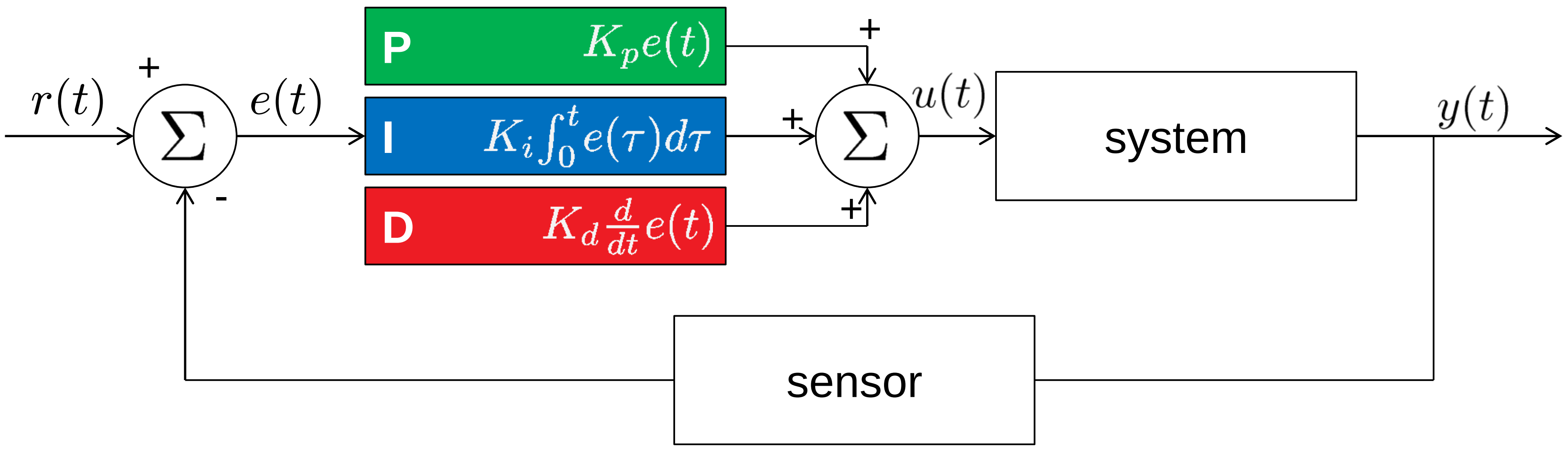
Control Theory / Control Systems Engineering

- mathematical/engineering discipline
- dealing with the understanding and controlling of the behavior of dynamical systems over time

A Single-Input-Single-Output (SISO) controller



Revision: PID Controller



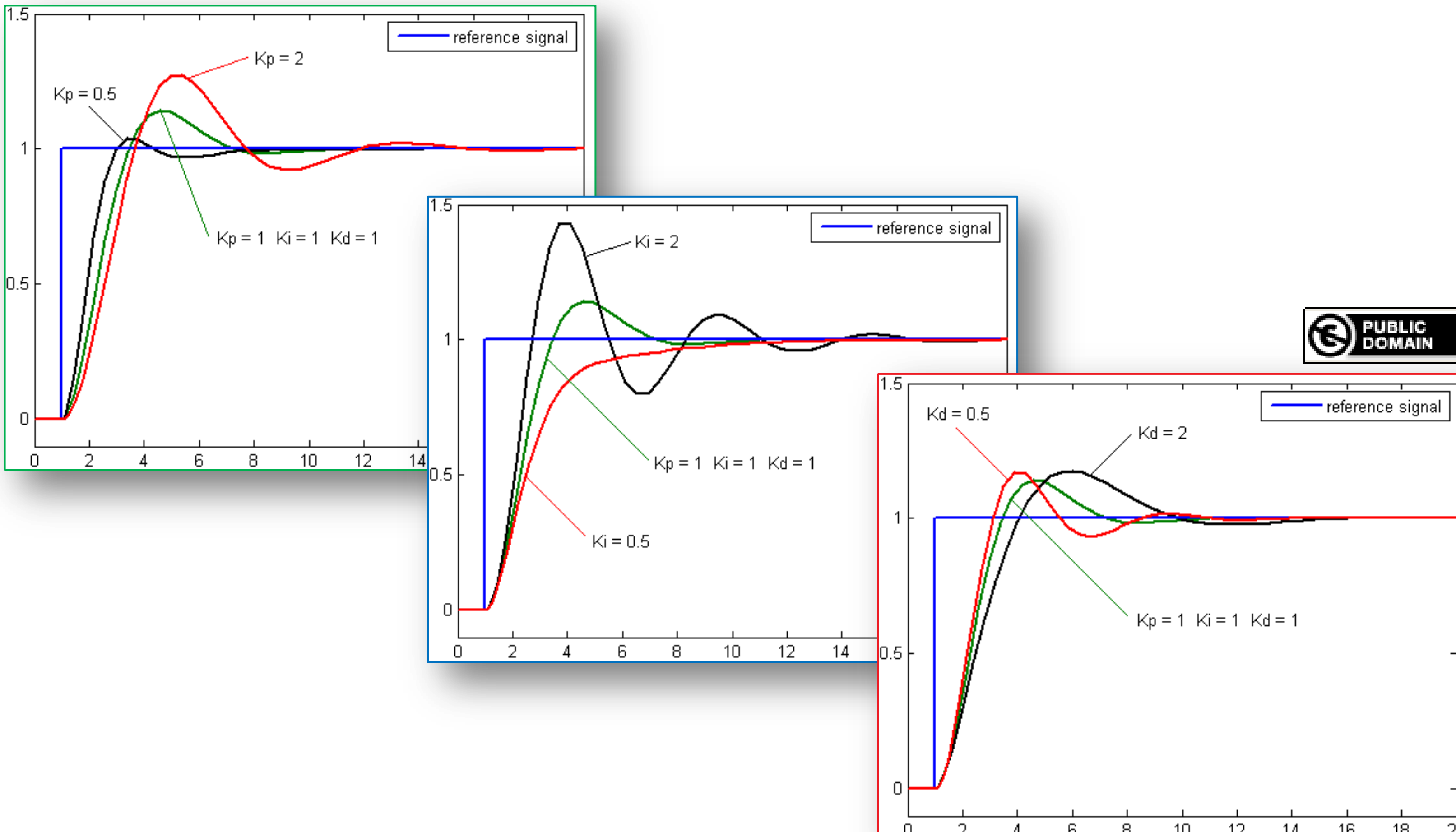
$$u(t) = \underbrace{K_p e(t)}_{\text{proportional part}} + \underbrace{K_i \int_0^t e(\tau)d\tau}_{\text{integral part}} + \underbrace{K_d \frac{d}{dt} e(t)}_{\text{derivative part}}$$

proportional part integral part derivative part

Influence of the Parameters

$$u(t) = \underbrace{K_p e(t)}_{\text{proportional part}} + \underbrace{K_i \int_0^t e(\tau) d\tau}_{\text{integral part}} + \underbrace{K_d \frac{d}{dt} e(t)}_{\text{derivative part}}$$

proportional part   integral part   derivative part



Link to Optimization

- the three variables K_p, K_i , and K_d have to be adjusted
- optimization: automated way of finding good solutions
  (other term: “parameter tuning”)

Online vs. Offline optimization

- offline = before deployment: finding the overall best system
- online = during deployment: finding the currently best response

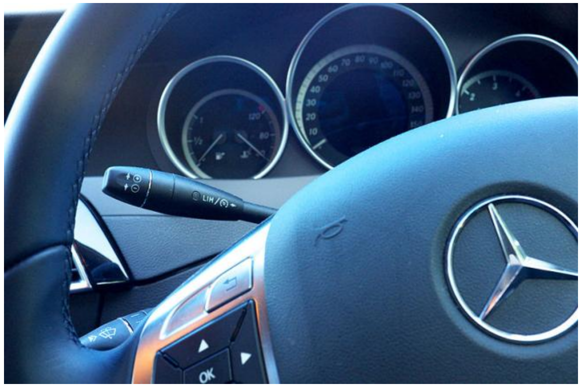
Deterministic vs. Stochastic/Randomized optimization

- deterministic = optimization result always determined by init. cond.
- random = use randomness to search

A Simple Control System Everybody Knows

Control Systems Engineering applied in several industrial sectors (keyword "embedded systems"), such as

- automotive sector, e.g. "cruise control"
- chemical engineering: "process control"
- robotics



Luigi Chiesa

Richard Greenhill / Hugo Elias

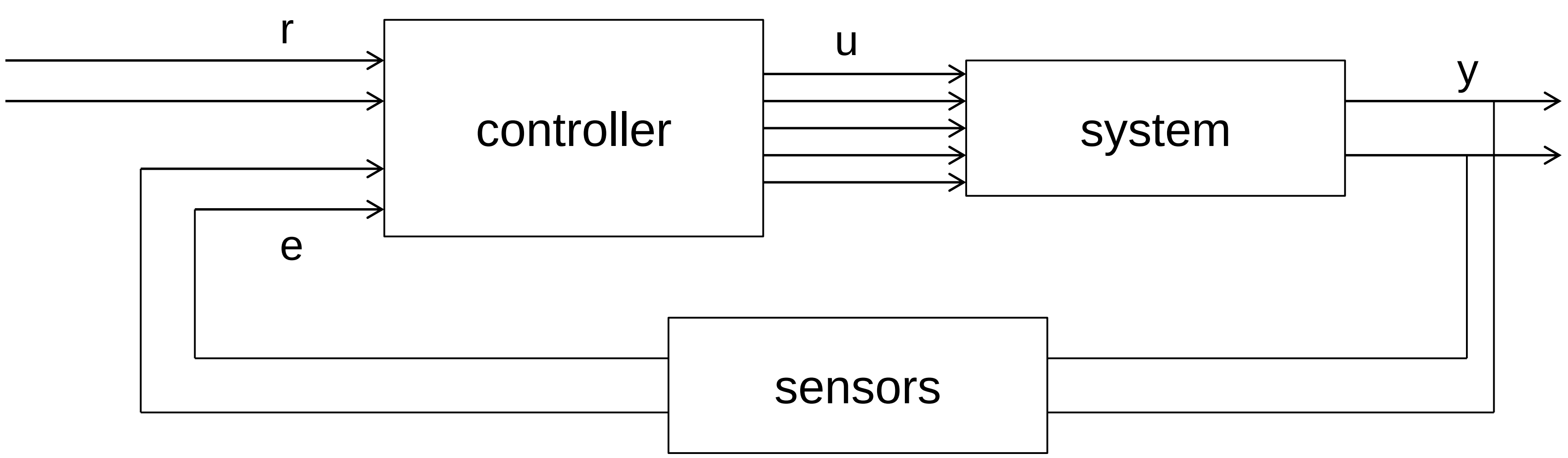
Advanced Control

- term not clearly defined
- other term could be "modern" control (in comparison to "classical" control theory)
- more complicated controllers
  - designed by means of meta-models
  - optimized by means of (randomized) search heuristics

Overview of the Course

Here:

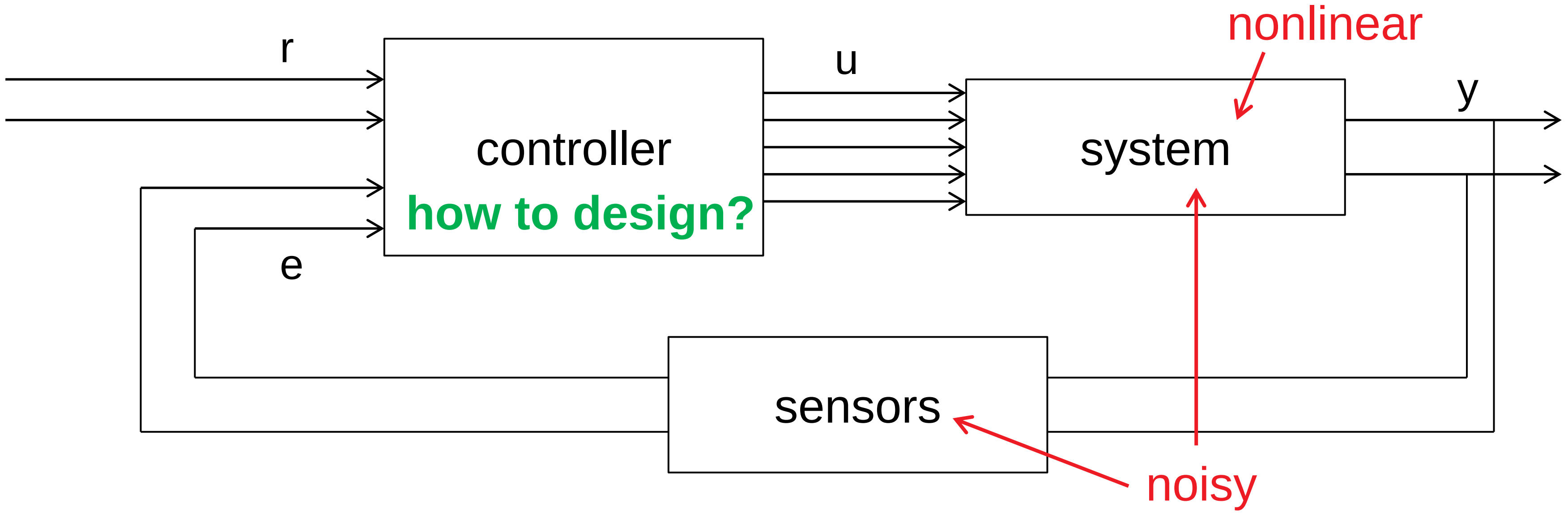
- multiple inputs, multiple outputs (MIMO) controllers
- nonlinear control
- noisy environments



Overview of the Course

Here:

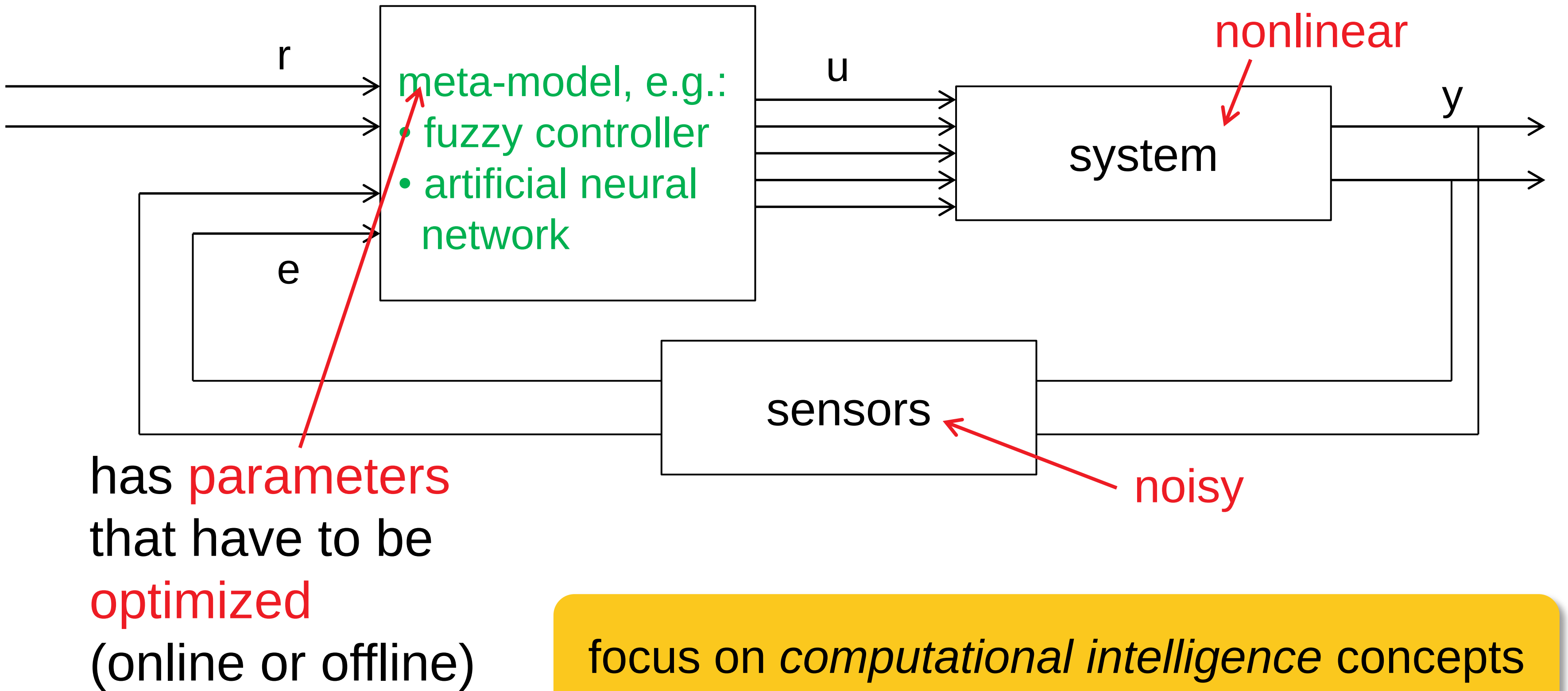
- multiple inputs, multiple outputs (MIMO) controllers
- nonlinear control
- noisy environments



Overview of the Course

Here:

- multiple inputs, multiple outputs (MIMO) controllers
- nonlinear control
- noisy environments



focus on *computational intelligence* concepts

Short Excursion: Computational Intelligence

Artificial Intelligence
- Computational Intelligence
  - Fuzzy Logic
  - Artificial Neural Networks (ANNs)
  - Evolutionary Computation (EC)
    - Genetic Algorithms (GAs)
    - Evolution Strategies (ESs)
    - Genetic Programming (GP)
    - Evolutionary Programming (EP)
    - Swarm Optimization
    - ...
- Machine Learning
- Robotics
- ...

Short Excursion: Computational Intelligence

Artificial Intelligence

- Computational Intelligence
  - Fuzzy Logic
  - Artificial Neural Networks (ANNs)
  - Evolutionary Computation (EC)
    - Genetic Algorithms (GAs)
    - Evolution Strategies (ESs)
    - Genetic Programming (GP)
    - Evolutionary Programming (EP)
    - Swarm Optimization
    - ...
- Machine Learning
- Robotics
- ...

focus of this course wrt. control

Course Overview

Date		Topic
Fri, 10.1.2014	DB	Introduction to Control, Examples of Advanced Control, Introduction to Fuzzy Logic
Fri, 17.1.2014	DB	Fuzzy Logic (cont'd), Introduction to Artificial Neural Networks
Fri, 24.1.2014	DB	Bio-inspired Optimization, discrete search spaces
Fri, 31.1.2014	AA	Continuous Optimization I
Fri, 7.2.2014	AA	Continuous Optimization II
break		
Fri, 28.2.2014	AA	The Traveling Salesperson Problem
Fr, 7.3.2014	DB	Controlling a Pole Cart
Fr, 14.3.2014		written exam (paper and computer)

all classes + exam at **8h00-11h15** (incl. a 15min break around 9h30) here in CTI-B3

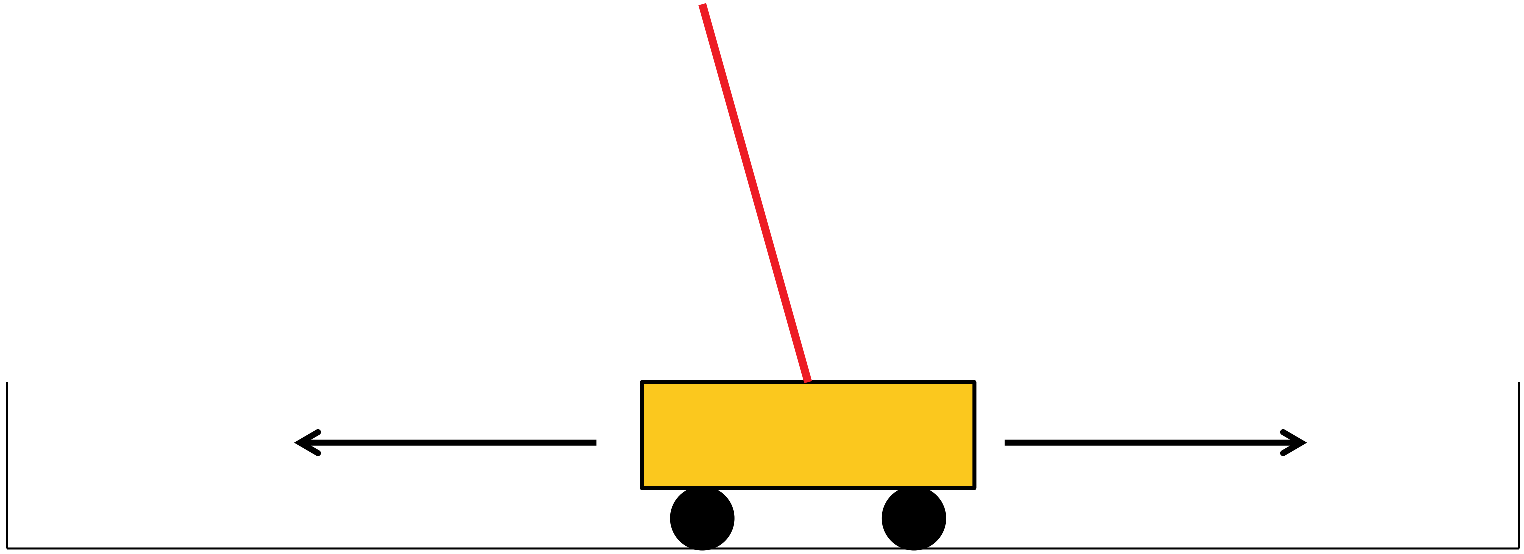
Advanced Control: An Example

Segway PT
introduced in 2001
20.1 km/h
ca. 1600 EUR

Simplified Example: The Pole Balancing Benchmark

Typical benchmark example of a system with "advanced control":

The Pole Balancing Problem

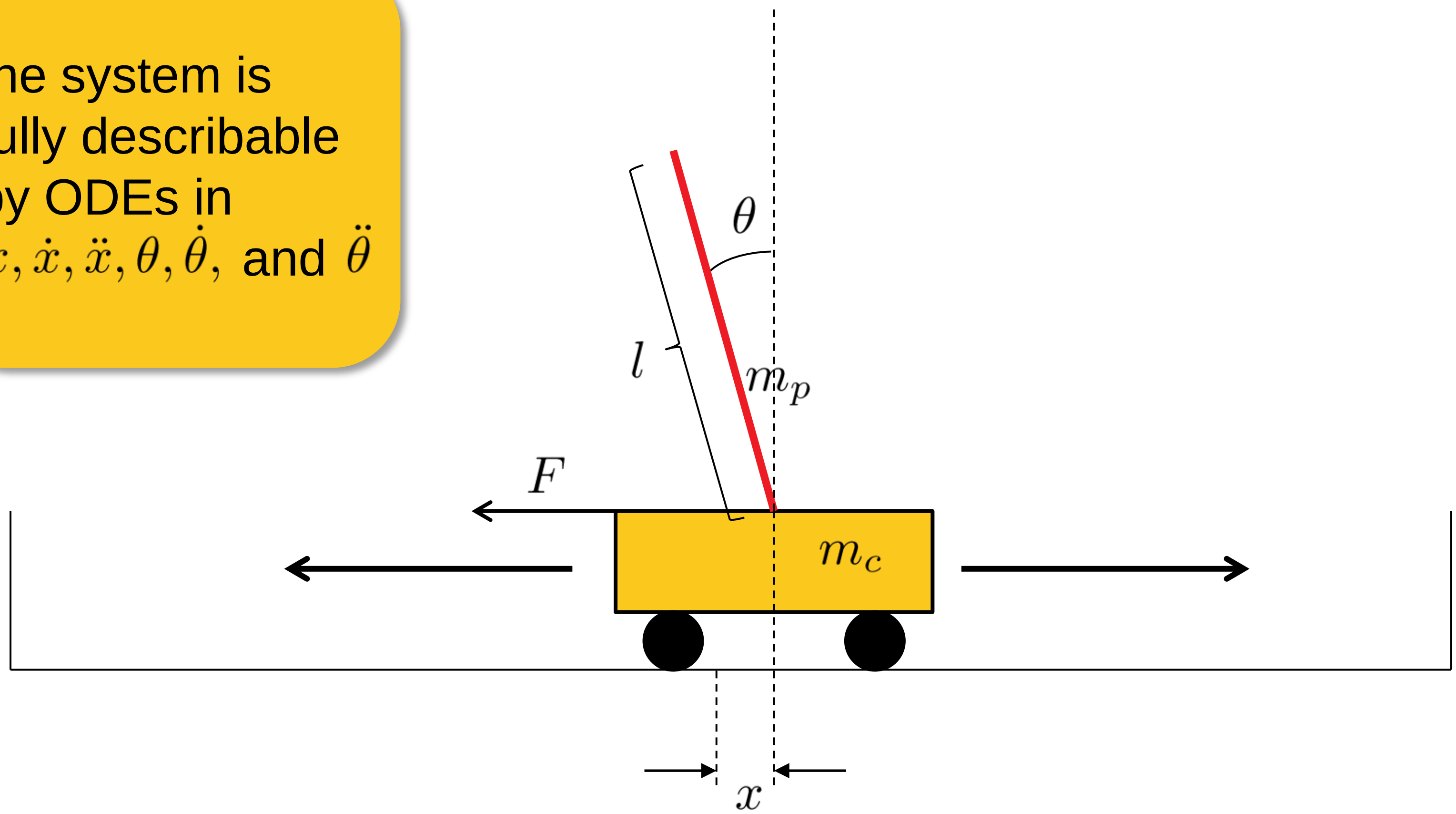


Simplified Example: The Pole Balancing Benchmark

Typical benchmark example of a system with "advanced control":

The Pole Balancing Problem

the system is fully describable by ODEs in $x, \dot{x}, \ddot{x}, \theta, \dot{\theta},$ and $\ddot{\theta}$



Simplified Example: The Pole Balancing Benchmark

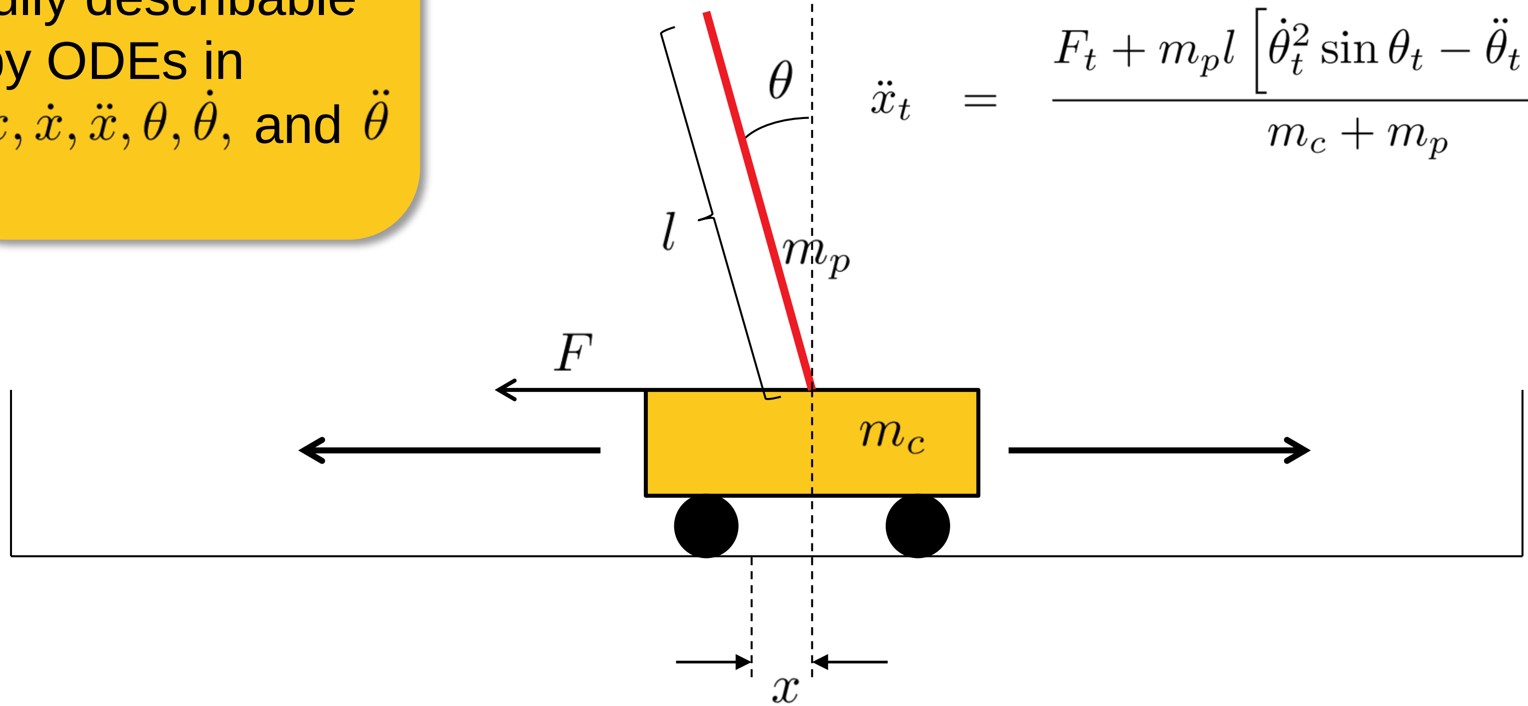
Typical benchmark example of a system with "advanced control":

The Pole Balancing Problem

the system is fully describable by ODEs in $x, \dot{x}, \ddot{x}, \theta, \dot{\theta}$, and $\ddot{\theta}$

$$\ddot{\theta}_t = \frac{g \sin\theta_t + \cos\theta_t \left[\frac{-F_t - m_p l \dot{\theta}_t^2 \sin\theta_t}{m_c + m_p}\right]}{l\left[\frac{4}{3} - \frac{m_p \cos^2\theta_t}{m_c + m_p}\right]}$$

$$\ddot{x}_t = \frac{F_t + m_p l \left[\dot{\theta}_t^2 \sin\theta_t - \ddot{\theta}_t \cos\theta_t\right]}{m_c + m_p}$$



http://researchers.lille.inria.fr/~brockhof/advancedcontrol/

$$\ddot{\theta}_t = \frac{g \sin\theta_t + \cos\theta_t \left[\frac{-F_t - m_p l \dot{\theta}_t^2 \sin\theta_t}{m_c + m_p}\right]}{l\left[\frac{4}{3} - \frac{m_p \cos^2\theta_t}{m_c + m_p}\right]} \qquad g \approx 9.81 m/s^2$$

$$\ddot{x}_t = \frac{F_t + m_p l \left[\dot{\theta}_t^2 \sin\theta_t - \ddot{\theta}_t \cos\theta_t\right]}{m_c + m_p}$$

Due to the constraints at all time steps t:

$$-r \leq \theta_t \leq +r$$

$$-h \leq x_t \leq +h$$

$$-F_{\max} \leq F_t \leq +F_{\max}$$

Simulated Pole Balancing

Given all the parameters of the system, what do we do with it?

Answer: simulate!

- starting point: certain (random) position and angle; velocities and accelerations are zero
- choose discretization time step (e.g. $\tau = 0.02s$)
- at each time step, do:
  - compute $\ddot{\theta}_t$ with values $\dot{\theta}_t$ and θ_t
  - compute $\ddot{x}_t$ with $\dot{\theta}_t, \theta_t$ and the new $\ddot{\theta}_t$
  - $$\begin{aligned} x_{t+1} &= x_t + \tau \dot{x}_t \\ \dot{x}_{t+1} &= \dot{x}_t + \tau \ddot{x}_t \\ \theta_{t+1} &= \theta_t + \tau \dot{\theta}_t \\ \dot{\theta}_{t+1} &= \dot{\theta}_t + \tau \ddot{\theta}_t \end{aligned}$$

! The above scheme is also known as Euler method

Polebalancing: Linear Control Law

Remark:

if the values and velocities of both position and angle are measured, there exists a linear (bang-bang) controller of the form:

$$F_t = F_m \operatorname{sgn}(k_1 x_t + k_2 \dot{x}_t + k_3 \theta_t + k_4 \dot{\theta}_t)$$

But still:

optimization necessary to estimate F_m, k_1, k_2, k_3, k_4

And what if

- not all sensors are available, or if they provide only noisy measurements?
- we take into account friction?
- the system shall work with different weights ("persons")?
- we have a more complicated problem (2D, 2 poles, ...)?

Exercise: Pole Balancing

http://researchers.lille.inria.fr/~brockhof/advancedcontrol/

Introduction to Fuzzy Logic

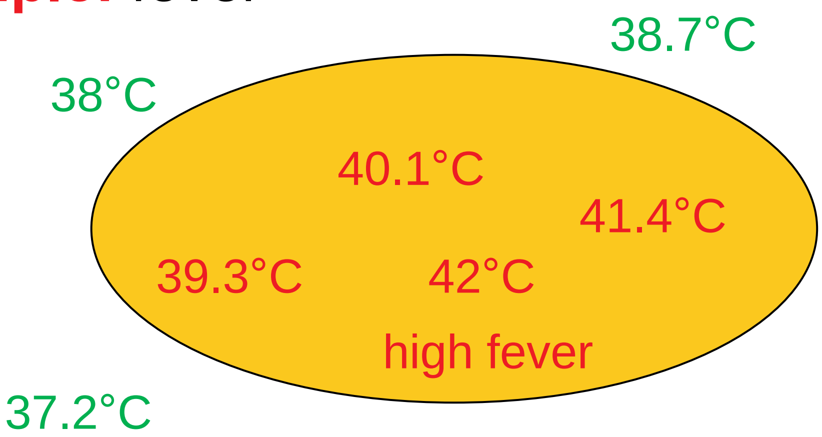
Fuzzy Logic

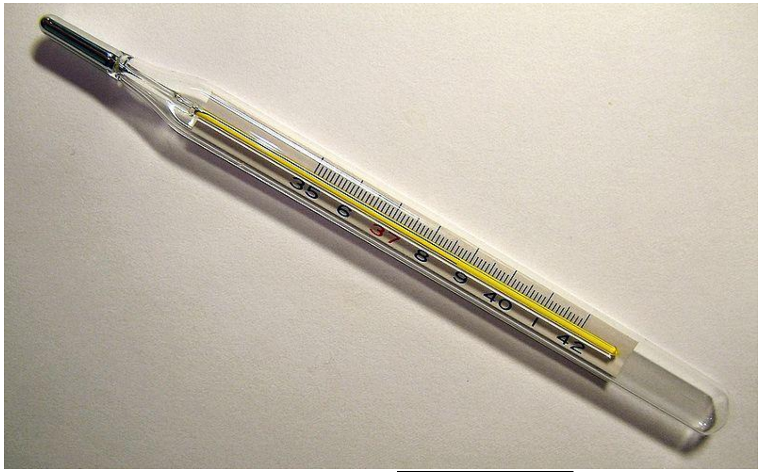
- introduced by Lotfi A. Zadeh at the University of California, Berkeley (*fuzzy sets* in 1965 and *fuzzy logic* in 1973)
- a mathematical tool to deal with uncertainties
- often described as "*computing with words*"[1]
  - e.g. {low, medium, high} instead of {0,1}
  - or "short" instead of "< 1 meter"

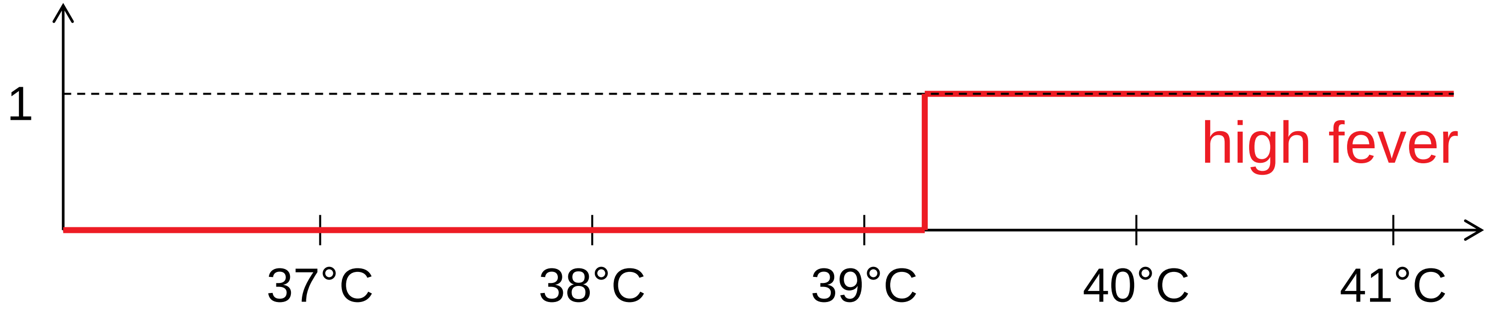
[1] L. A. Zadeh: Fuzzy logic = computing with words. In IEEE Transactions on Fuzzy Systems, 4(2), p. 103-111. 1996

- standard sets: either a in A or a not in A
- fuzzy sets: a in A with probability p_a

Example: fever







- standard sets: either a in A or a not in A
- fuzzy sets: a in A with probability p_a

Example: fever

38.7°C

38°C

40.1°C

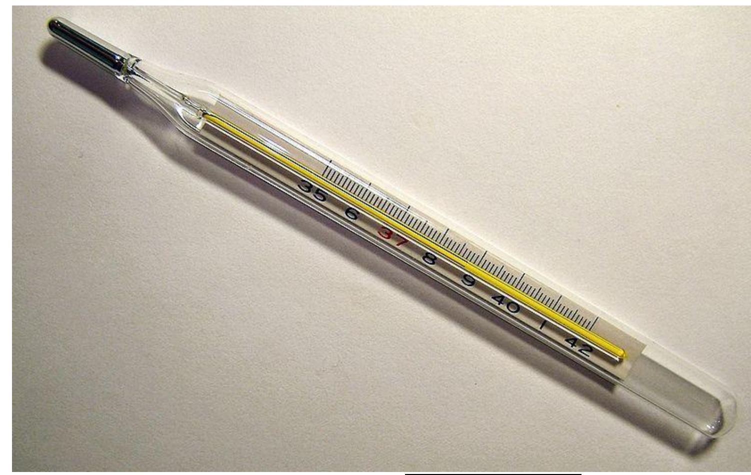
41.4°C

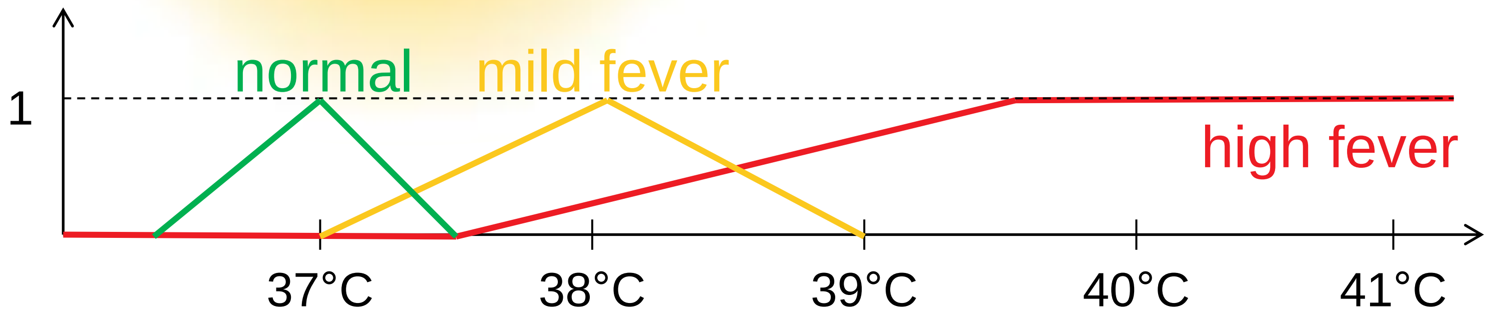
39.3°C

42°C

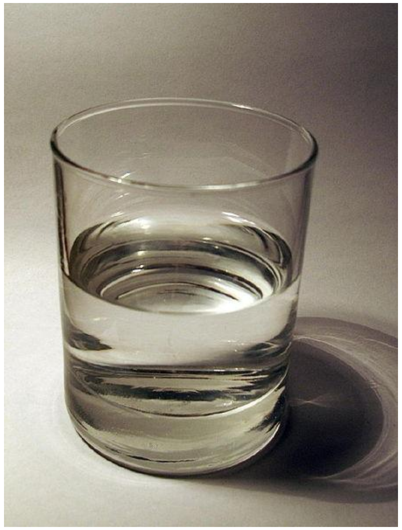
high fever

37.2°C



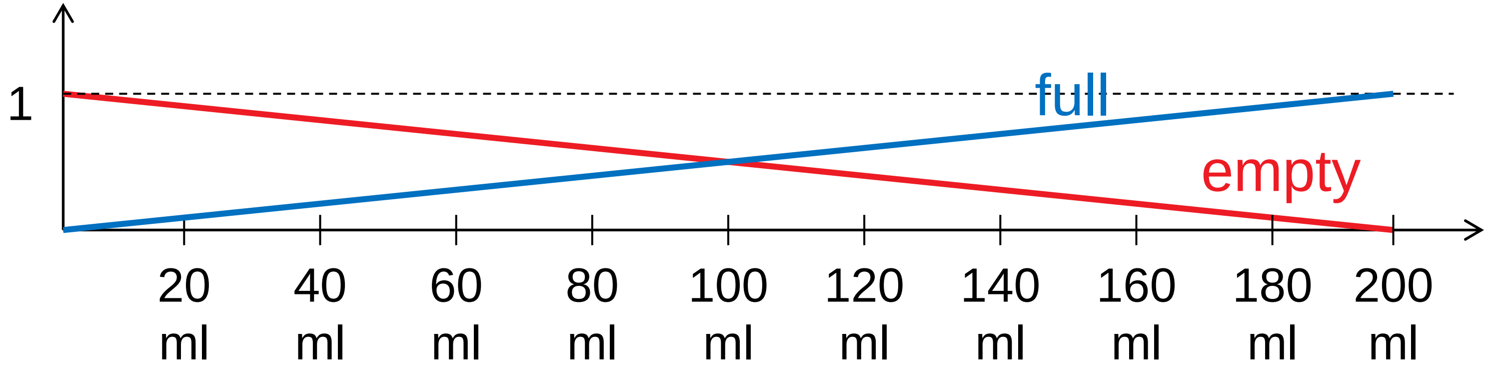


Idea of Membership Function

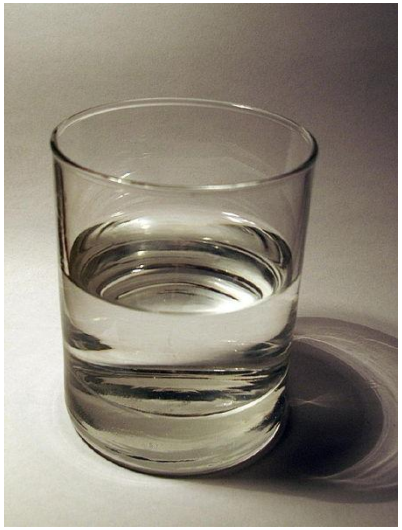


Jaques

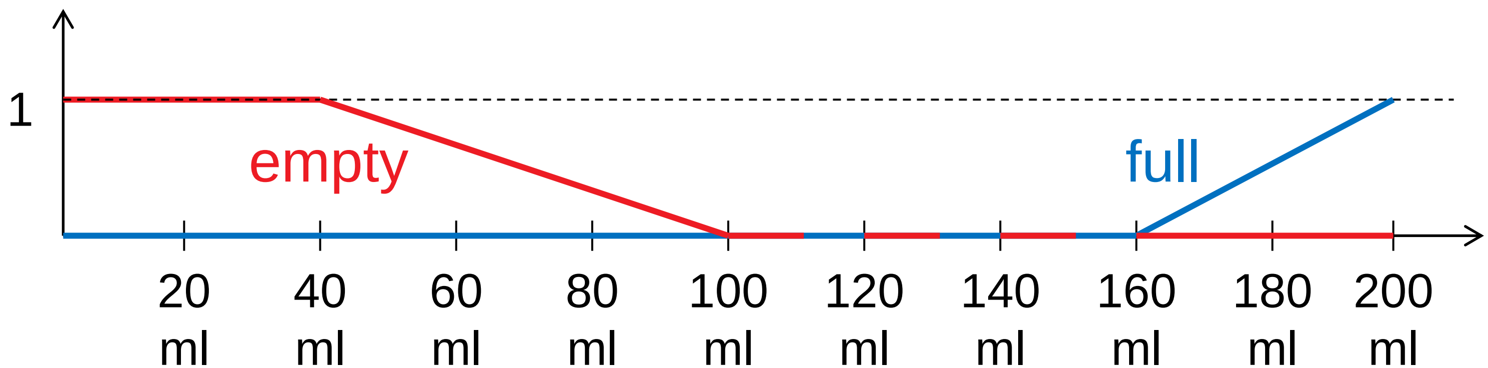
- 200ml glass with 100ml water: full or empty?
- standard logic: either full or empty
- fuzzy logic: glass can be full and empty!
  - 100ml: glass 50% full and 50% empty
  - 40ml: glass 20% full and 80% empty
  - but also more complex membership functions possible!



Idea of Membership Function

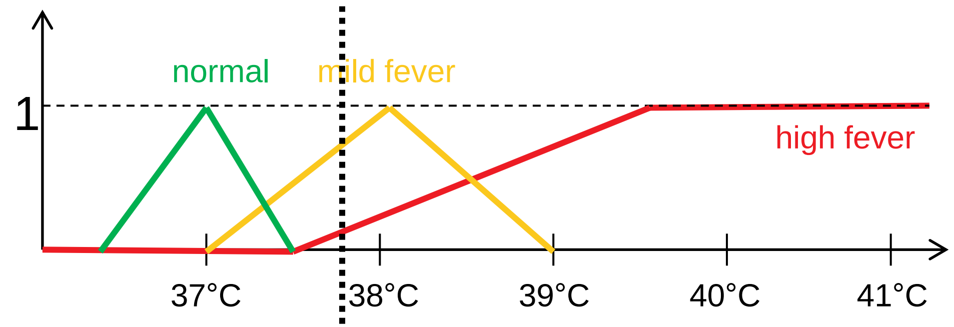

Jaques

- 200ml glass with 100ml water: full or empty?
- standard logic: either full or empty
- fuzzy logic: glass can be full and empty!
  - 100ml: glass 50% full and 50% empty
  - 40ml: glass 20% full and 80% empty
  - but also more complex membership functions possible!



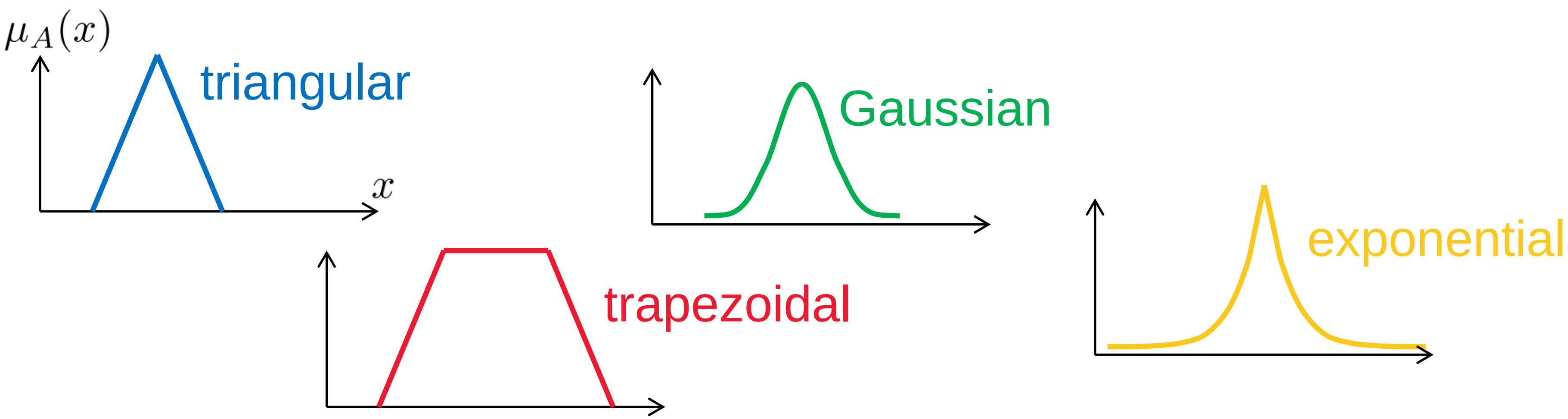
Fuzzification:

= transferring a real-valued variable into a fuzzy one



80% mild and 10% high fever

Several membership functions $\mu_A(x)$ known to do that:



In the end...

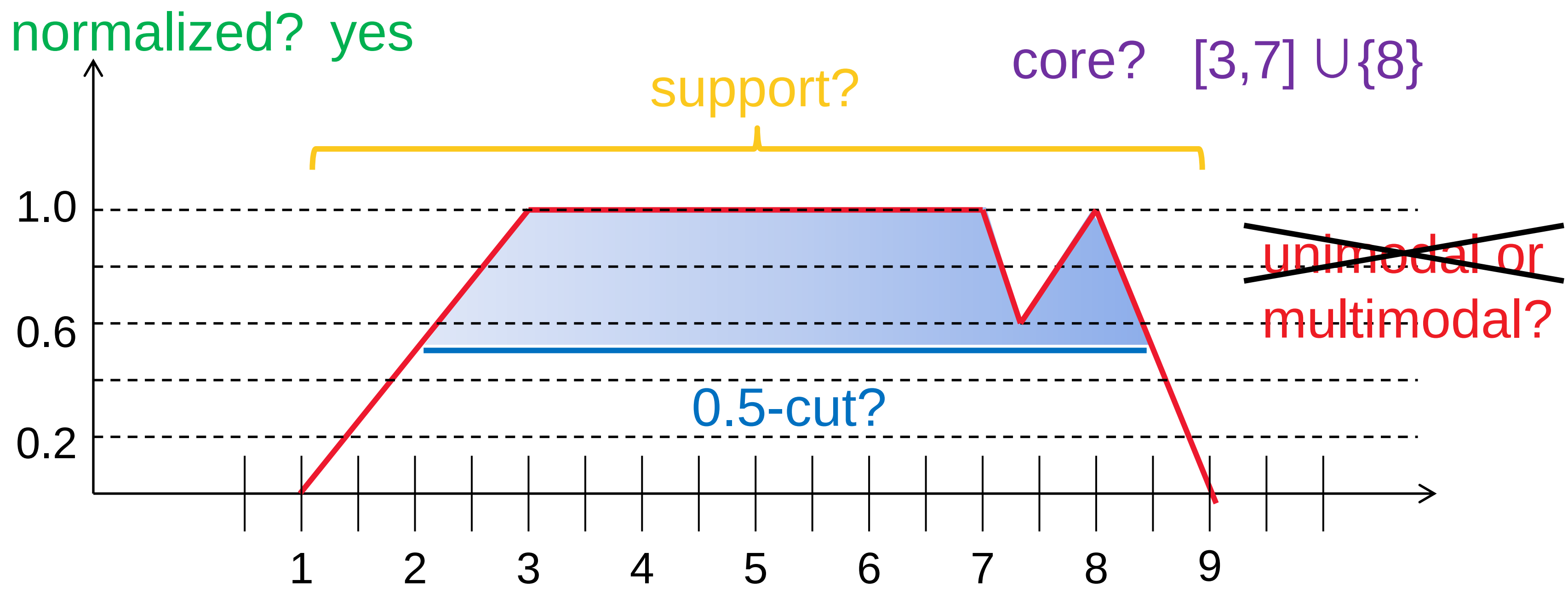
...everything is based on intuition (there are no strict rules)

Properties of Membership Functions

- μ_A is called normalized if its height is 1
- $\{x \,|\, \mu_A(x) > 0\}$ is called the support of μ_A
- $\{x \,|\, \mu_A(x) = 1\}$ is called the core of μ_A
- An α-cut of μ_A is the set $A_\alpha = \{x \,|\, \mu_A(x) \geq \alpha\}$
- If μ_A contains only one maximum, we call μ_A unimodal and A convex
- otherwise, μ_A is called multimodal and A nonconvex

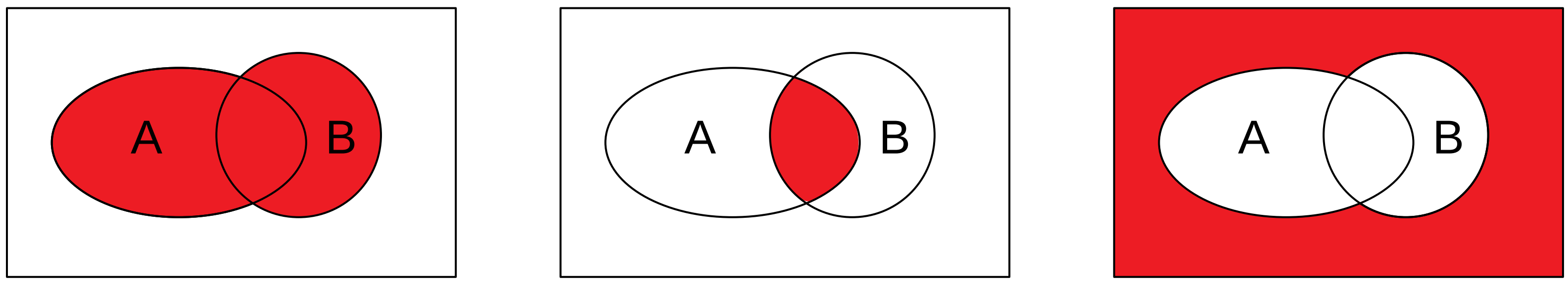
Properties of Membership Functions

- μ_A is called normalized if its height is 1
- $\{x \mid \mu_A(x) > 0\}$ is called the support of μ_A
- $\{x \mid \mu_A(x) = 1\}$ is called the core of μ_A
- An α-cut of μ_A is the set $A_\alpha = \{x \mid \mu_A(x) \geq \alpha\}$
- If μ_A contains only one maximum, we call μ_A unimodal and A convex
- otherwise, μ_A is called multimodal and A nonconvex

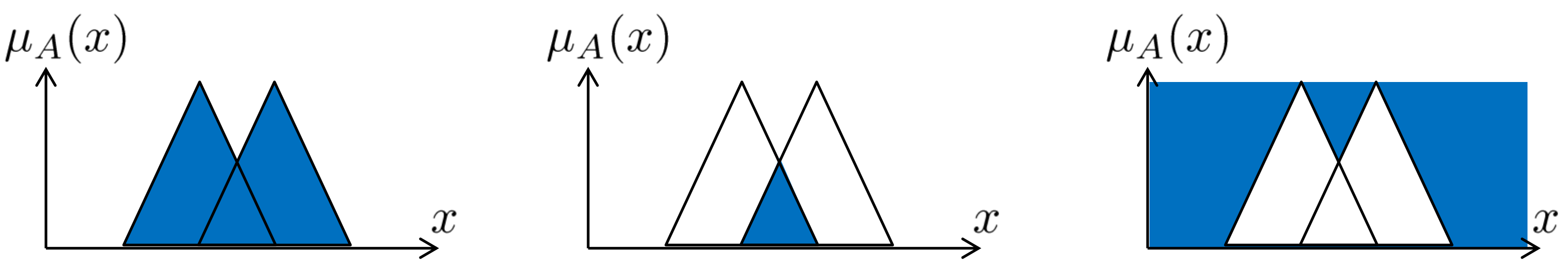


Union, intersection, and complement:

standard logic



fuzzy logic

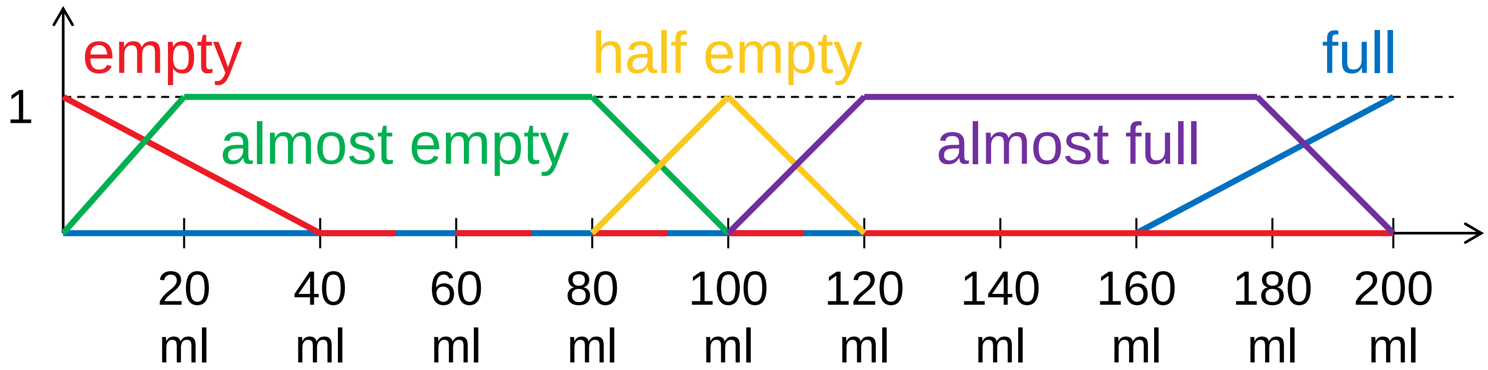


union = max

intersection = min

complement = 1-x

Defuzzifying



How do we get back “crisp” numbers (fuzzy set → real number)?

- there are many ways of doing it!

Maximum defuzzification: take x^* with $\forall x: \mu_A(x^*) \geq \mu_A(x)$

- simple but not accurate if μ_A multimodal

Centroid defuzzification: $x^* = \frac{\int \mu_A(x) x dx}{\int \mu_A(x) dx}$

- very accurate
- might be complicated to compute
- often used

Fuzzy Logic: Inferring Statements

Classical Logic:

- IF p THEN q
- equivalent to $\neg p \vee q$

	q = true	q = false
p = true	true	false
p = false	true	true

Fuzzy Logic:

- not so easy with fuzzy sets
  - interpretation as $\neg p \vee q$ results in some undesired effects
  - hence, rather "inference" than implication (for math. reasons)
- in general, implication is a function $\mu(x, y) = \Phi(\mu_A(x), \mu_B(y))$
- > 40 different implication rules proposed
- here, we consider only three (the easy and most used ones)

Fuzzy Logic: Inferring Statements

The sharp implication:

- $\mu(x,y) = \Phi(\mu_A(x), \mu_B(y)) = \begin{cases} 1 & \text{if } \mu_A(x) \leq \mu_B(y) \\ 0 & \text{else} \end{cases}$
- intuition: if X and Y are crisp sets, then $X \Rightarrow Y$ iff $X \subseteq Y$

	q=0	q=0.5	q=1
p=0	1	1	1
p=0.5	0	1	1
p=1	0	0	1

Mamdani's inference[1]:

membership function of implication:

$$\mu(x,y) = \Phi(\mu_A(x), \mu_B(y)) = \min(\mu_A(x), \mu_B(y))$$

only ¼ of corner values equal to 2-valued logic!

inference, no implication

	q=0	q=0.5	q=1
p=0	0	0	0
p=0.5	0	0.5	0.5
p=1	0	0.5	1

[1] E. H. Mamdani. "Application of fuzzy logic to approximate reasoning using linguistic synthesis". IEEE Transactions on Computers, C-26(12):1182–1191, December 1977.

Fuzzy Logic: Inferring Statements

Larsen Product implication[1]**:**

membership function of implication:

$$\mu(x,y) = \Phi(\mu_A(x), \mu_B(y)) = \mu_A(x) \cdot \mu_B(y)$$

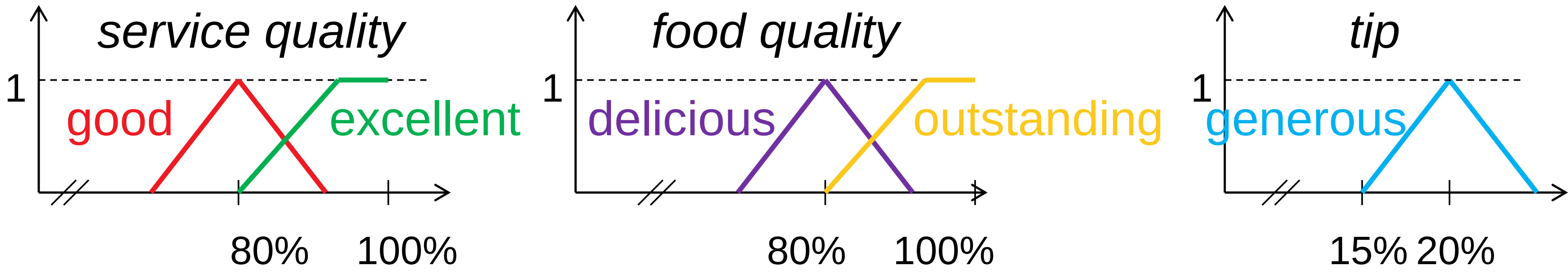
again: only ¼ of corner values equal 2-valued logic!

inference, no implication

	q=0	q=0.5	q=1
p=0	0	0	0
p=0.5	0	0.25	0.5
p=1	0	0.5	1

[1] P. M. Larsen, "Industrial Applications of Fuzzy Logic Control", International Journal of Man-Machine Studies, Vol. 12, No. 1, 1980, pp. 3-10.

Example

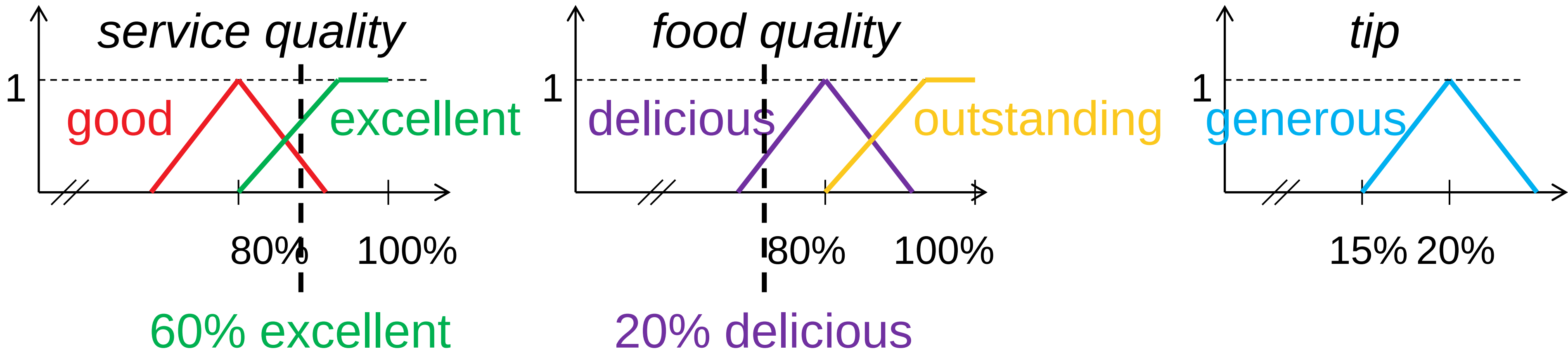


IF service is excellent AND food is delicious THEN tip is generous

What happens for different service and food qualities?

1. fuzzify inputs
2. compute value of left-hand side
3. then apply above rule (e.g. wrt. Mamdani's rule)
4. use defuzzification rule (e.g. centroid)

Example



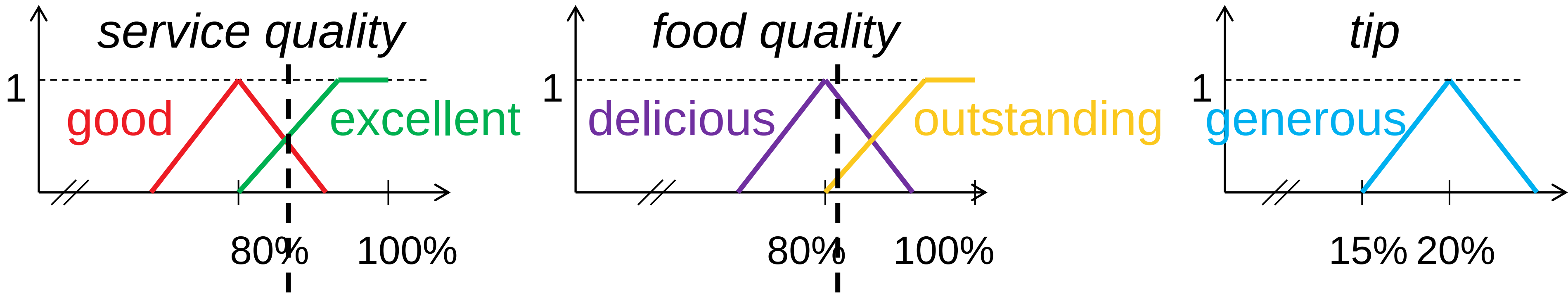
IF service is excellent AND food is delicious THEN tip is generous

What happens for different service and food qualities?

➊ fuzzify:

- 60% excellent AND 20% delicious

Example



50% excellent

90% delicious

IF service is excellent AND food is delicious THEN tip is generous

What happens for different service and food qualities?

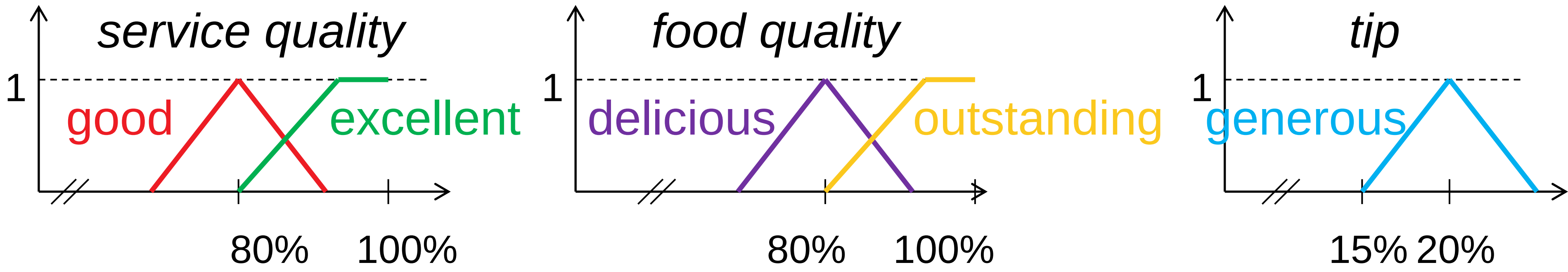
❶ fuzzify:

- 60% excellent AND 20% delicious
- 50% excellent AND 90% delicious

❷ compute value of left-hand: here “AND = min.”

- 20%
- 50%

Example

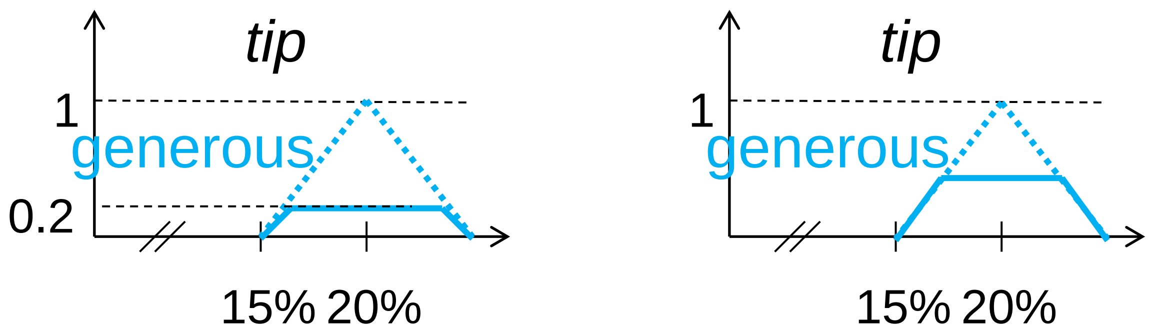


IF service is excellent AND food is delicious THEN tip is generous

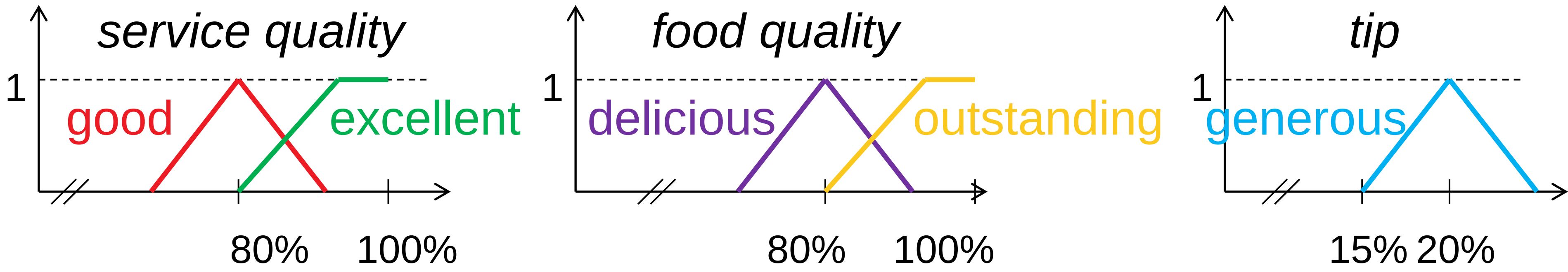
What happens for different service and food qualities?

❸ apply Mamdani's rule: $\mu(x,y) = \min(\mu_A(x), \mu_B(y))$

$\mu_A(x) = 20\%$ $\mu_A(x) = 50\%$



Example

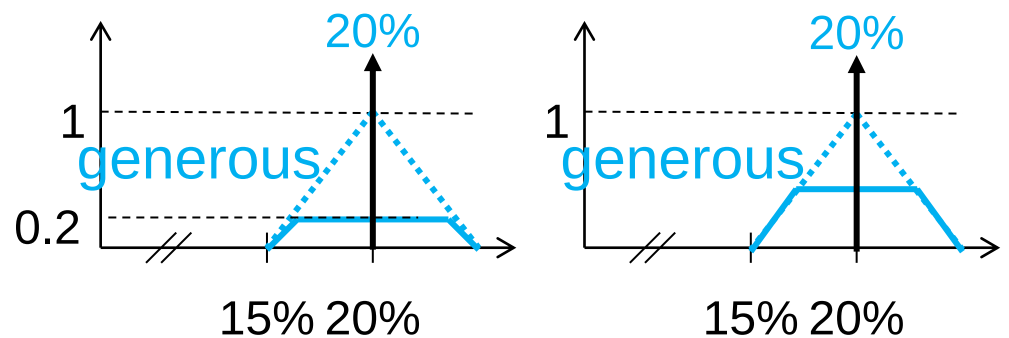


IF service is excellent AND food is delicious THEN tip is generous

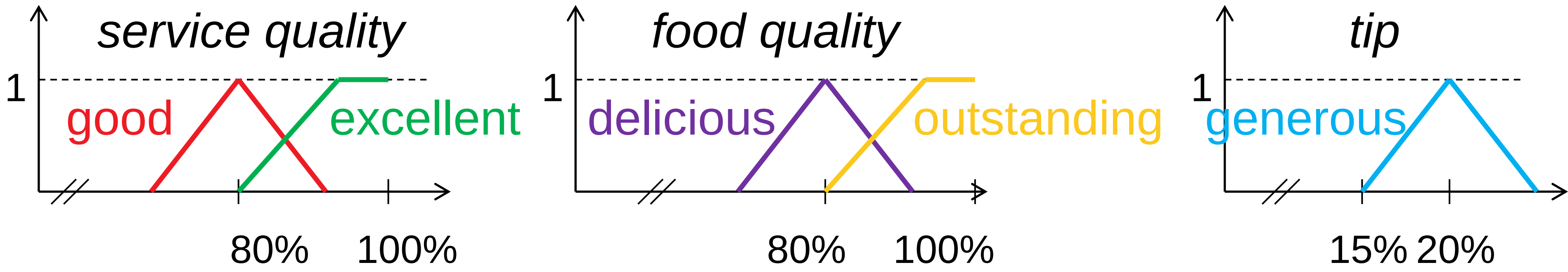
What happens for different service and food qualities?

❹ use defuzzification rule (e.g. centroid)

here: same result, but also only 1 rule applied



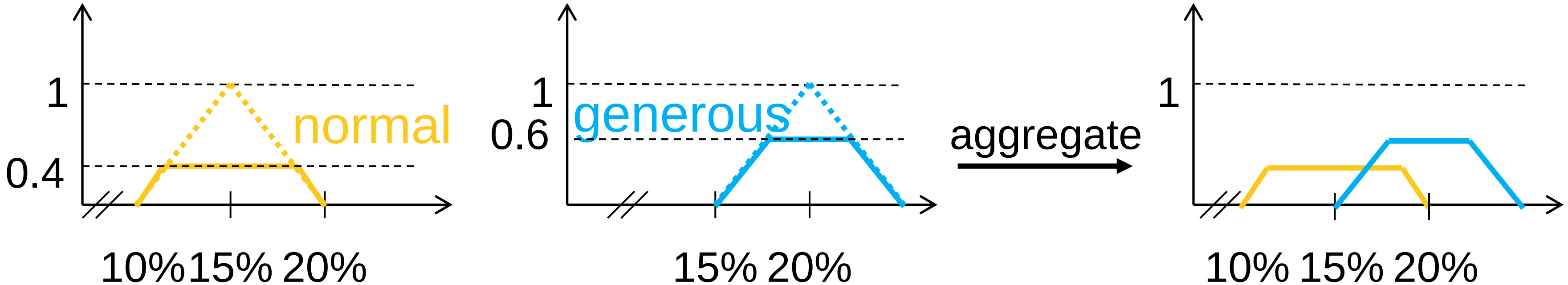
Example



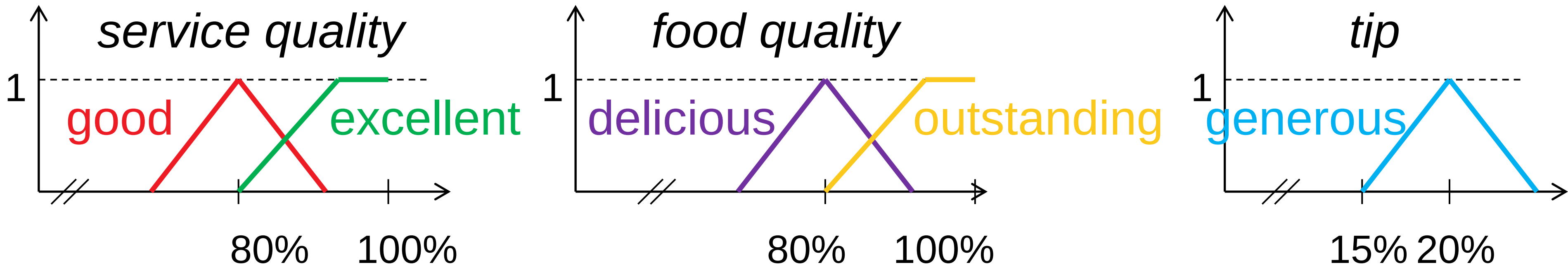
IF service is normal AND food is normal THEN tip is normal
IF service is excellent AND food is delicious THEN tip is generous

Multiple rules

- a) apply all inference rules
- b) aggregate resulting membership functions (e.g. with max.)



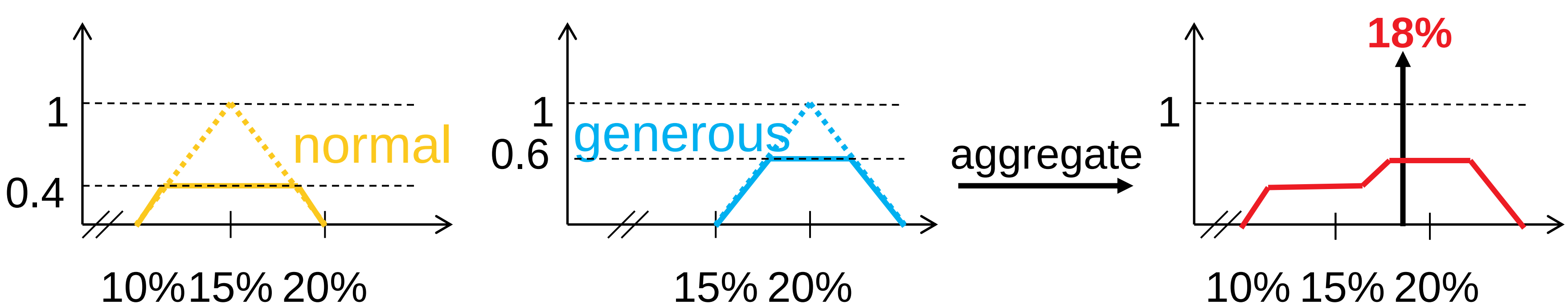
Example



IF service is normal AND food is normal THEN tip is normal
IF service is excellent AND food is delicious THEN tip is generous

Multiple rules

- ❸ a) apply all inference rules
- ❸ b) aggregate resulting membership functions (e.g. with max.)



Fuzzy Control

"Classical" control:

- mathematical ("crisp") formulations
- based on mathematical models, especially ODEs
- e.g. "210°C < TEMP < 220°C"

Fuzzy control:

- design formalized by words
- based on experience of the designer
- e.g. "IF (process is too cool) AND (process is getting colder) THEN (add heat to the process)" or "IF (process is too hot) AND (process is heating rapidly) THEN (cool the process quickly)"

A Simple Rule Matrix

Back to the water tap problem from last week:

- imagine measurements of temperature and water flow (e.g. per second) and the controllable inputs "hot water" and "cold water"
- further assume the inputs are fuzzified as {too cold, fine, too hot} (for the temperature) and {not enough, fine, too much} (for the water flow)

Then, a 3x3 rule matrix can show the responses:

	too cold	fine	too hot
not enough			
fine			
too much			

A Simple Rule Matrix

Back to the water tap problem from last week:

- imagine measurements of temperature and water flow (e.g. per second) and the controllable inputs “hot water” and “cold water”
- further assume the inputs are fuzzified as {too cold, fine, too hot} (for the temperature) and {not enough, fine, too much} (for the water flow)

Then, a 3x3 rule matrix can show the responses:

	too cold	fine	too hot
not enough	increase hot	increase hot & cold	increase cold
fine	decrease cold & increase hot	do nothing	increase cold & decrease hot
too much	decrease cold	decrease hot & cold	decrease hot

e.g. IF temperature is fine AND water flow is not enough THEN increase both cold and hot water

Another Rule Matrix

Example: electric heater

- given: goal temperature T_{opt}
- measured: temperature T and temperature change dT/dt
- controlled inputs: heat (heating on) and cool (fan on)
- fuzzify: $T-T_{opt}$ and $d(T-T_{opt})/dt$ in {negative, zero, positive}

		temperature: $T-T_{opt}$		
		negative	zero	positive
temperature change: $d(T-T_{opt})/dt$	negative			
	zero			
	positive			

Another Rule Matrix

Example: electric heater

- given: goal temperature T_{opt}
- measured: temperature T and temperature change dT/dt
- controlled inputs: heat (heating on) and cool (fan on)
- fuzzify: $T-T_{opt}$ and $d(T-T_{opt})/dt$ in {negative, zero, positive}

		temperature: $T-T_{opt}$		
		negative	**zero**	**positive**
temperature change: $d(T-T_{opt})/dt$	**negative**	heat	heat	cool
	zero	heat	do nothing	cool
	positive	heat	cool	cool

Remarks on Rule Matrices

- nothing fancy, but assisting to not forget a rule
- not much helpful if >2 input variables
- not always necessary to define output for all input combinations
- not usable if rules are not of the form “IF a AND b THEN c”
- odd number of rows and columns often helpful (to have a “zero” state with no change)

Again: What if a fuzzified “crisp” input value fire >1 rule?

- then: aggregation (union, max) of output membership functions

How to Design a Fuzzy Controller

1) Define control objectives and criteria

   What am I trying to control? What do I have to do to control the system? What kind of response do I need? What are the possible (probable) system failure modes?

2) Determine input/output relationships and choose the variables.
3) Break the control problem down into a series of IF X AND Y THEN Z rules (or similar) that define the desired system output response for given system input conditions.

   ! If possible, use at least one variable and its time derivative.

4) Create Fuzzy Logic membership functions and decide on inference rules that define the meaning (values) of the Input/Output terms used in your rules.
5) Implement the system in software (or hardware).
6) Test, evaluate, and tune the rules and membership functions, until satisfactory results are obtained.

according to the Fuzzy Logic Tutorial by Steven D. Kaehler
http://www.seattlerobotics.org/encoder/mar98/fuz/flindex.html

Exercise:
A Fuzzy Controller for the Pole Balancing Problem

Conclusions

I hope it became clear...

...what advanced control is about

...what the pole balancing problem is

...what a fuzzy control system is

...and that designing a good controller is not always easy